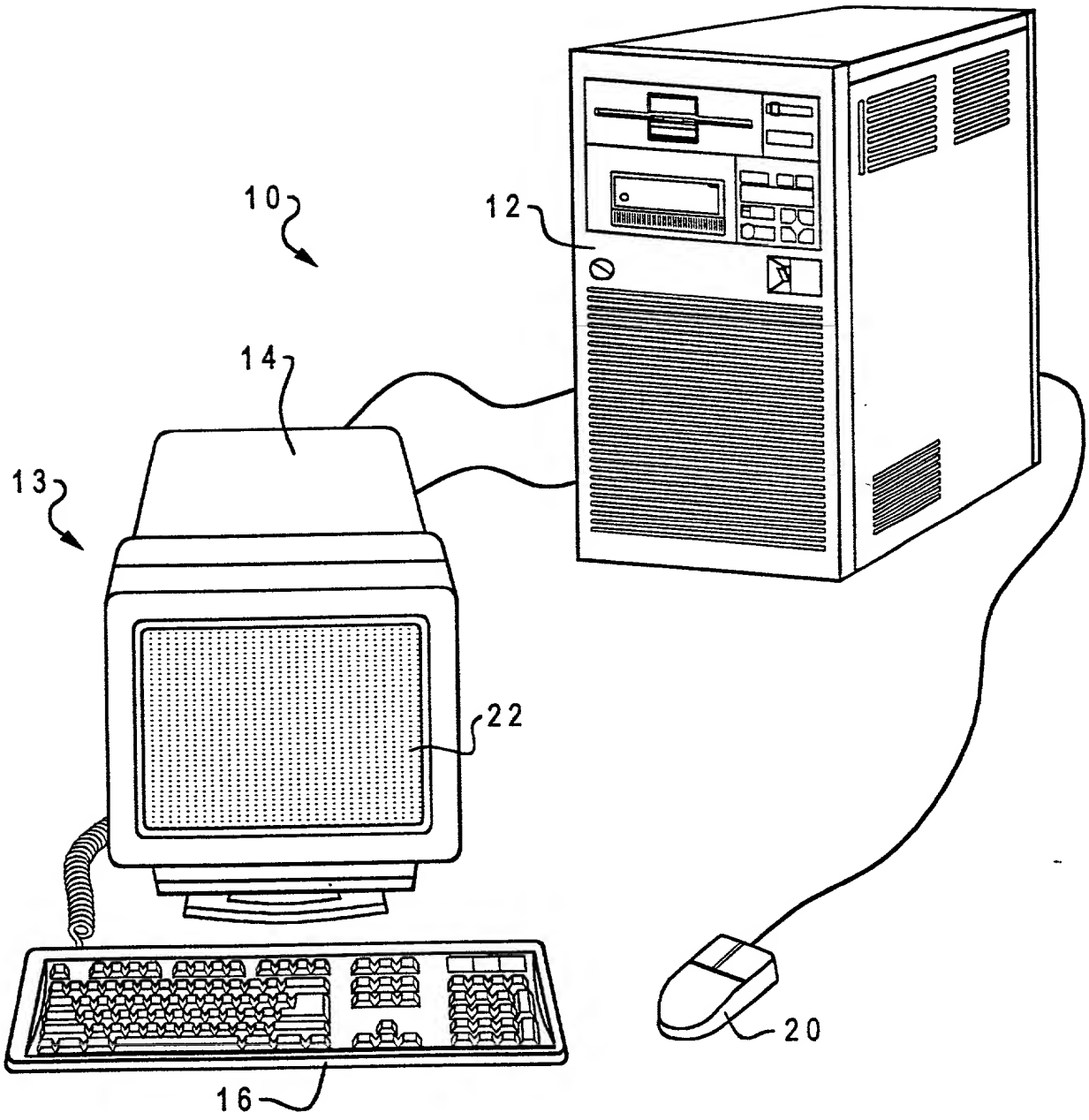

Fig. 1

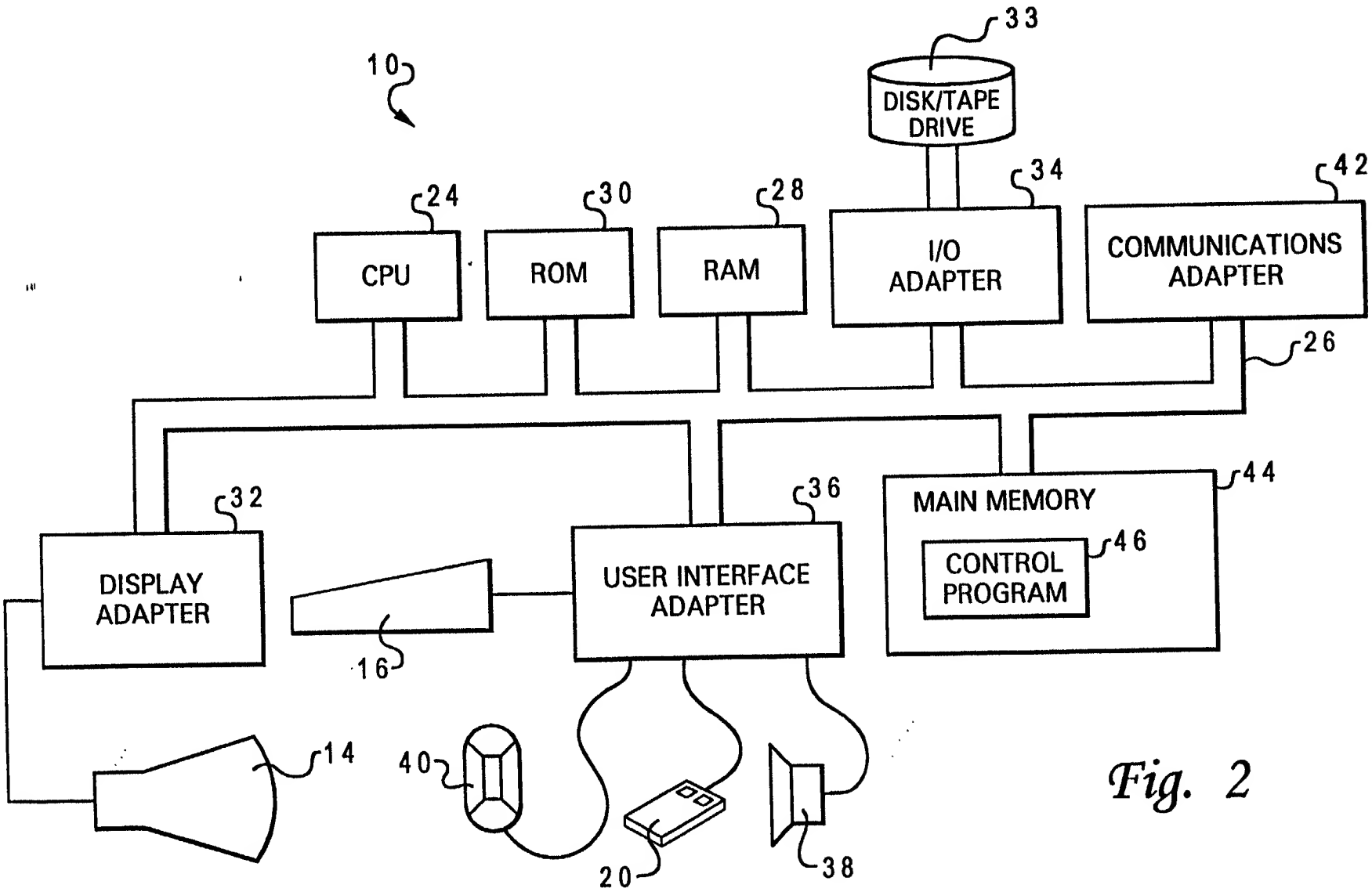

Fig. 2

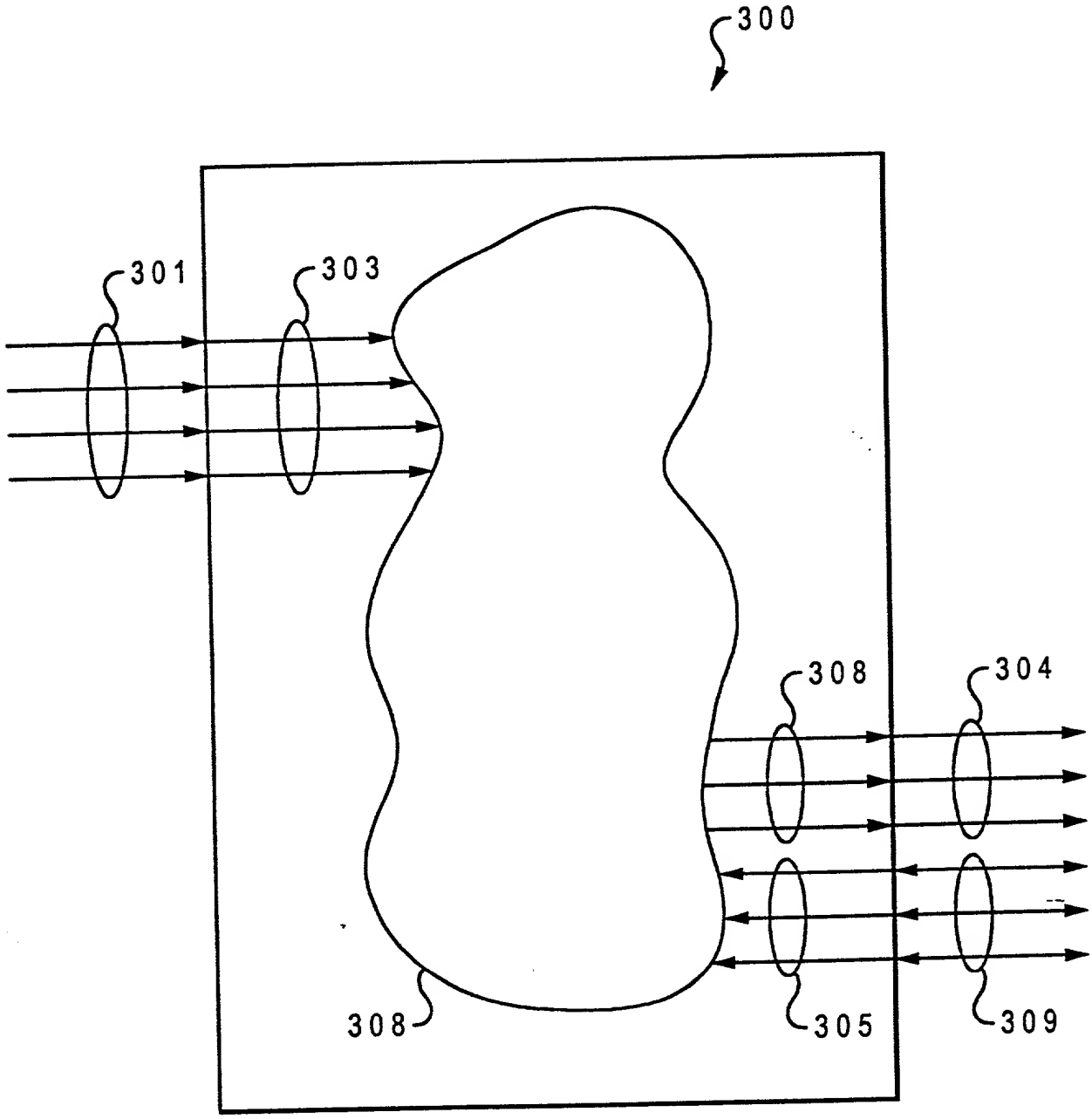

Fig. 3A

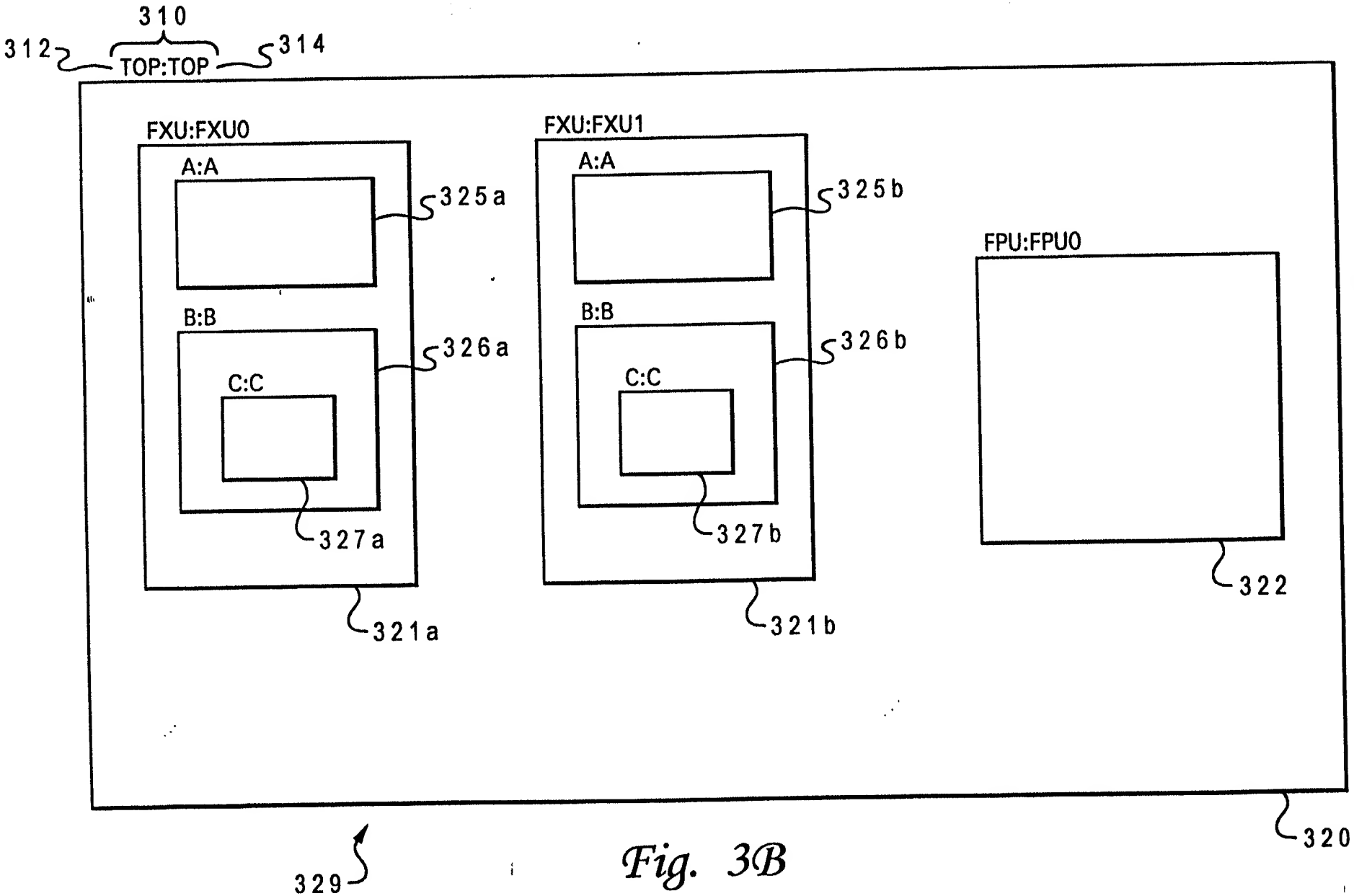

Fig. 3B

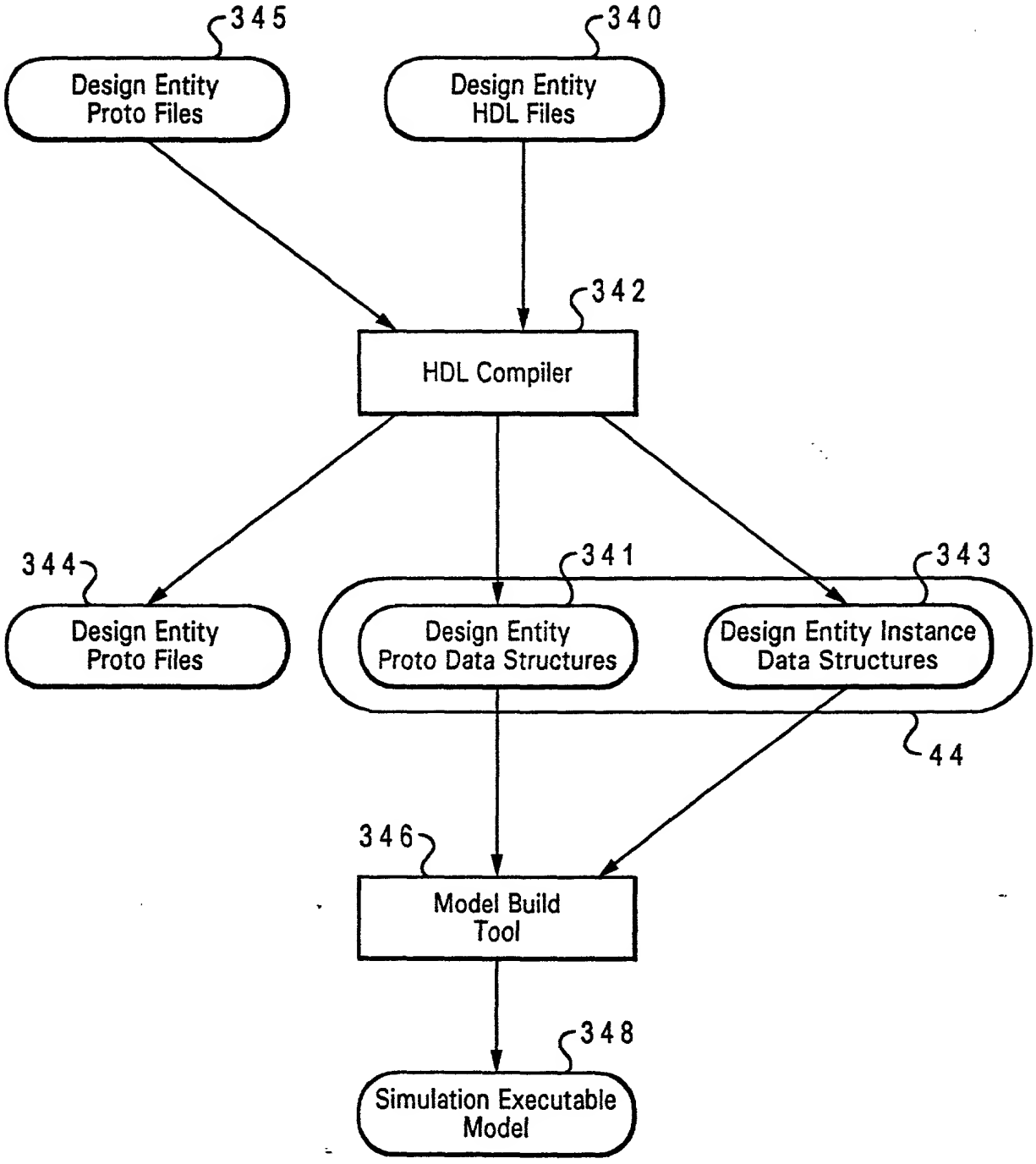
345
Design Entity
Proto Files
340
Design Entity
HDL Files
342
HDL Compiler
344
Design Entity
Proto Files
341
Design Entity
Proto Data Structures
343
Design Entity Instance
Data Structures
44
346
Model Build
Tool
348
Simulation Executable
Model

Fig. 3C

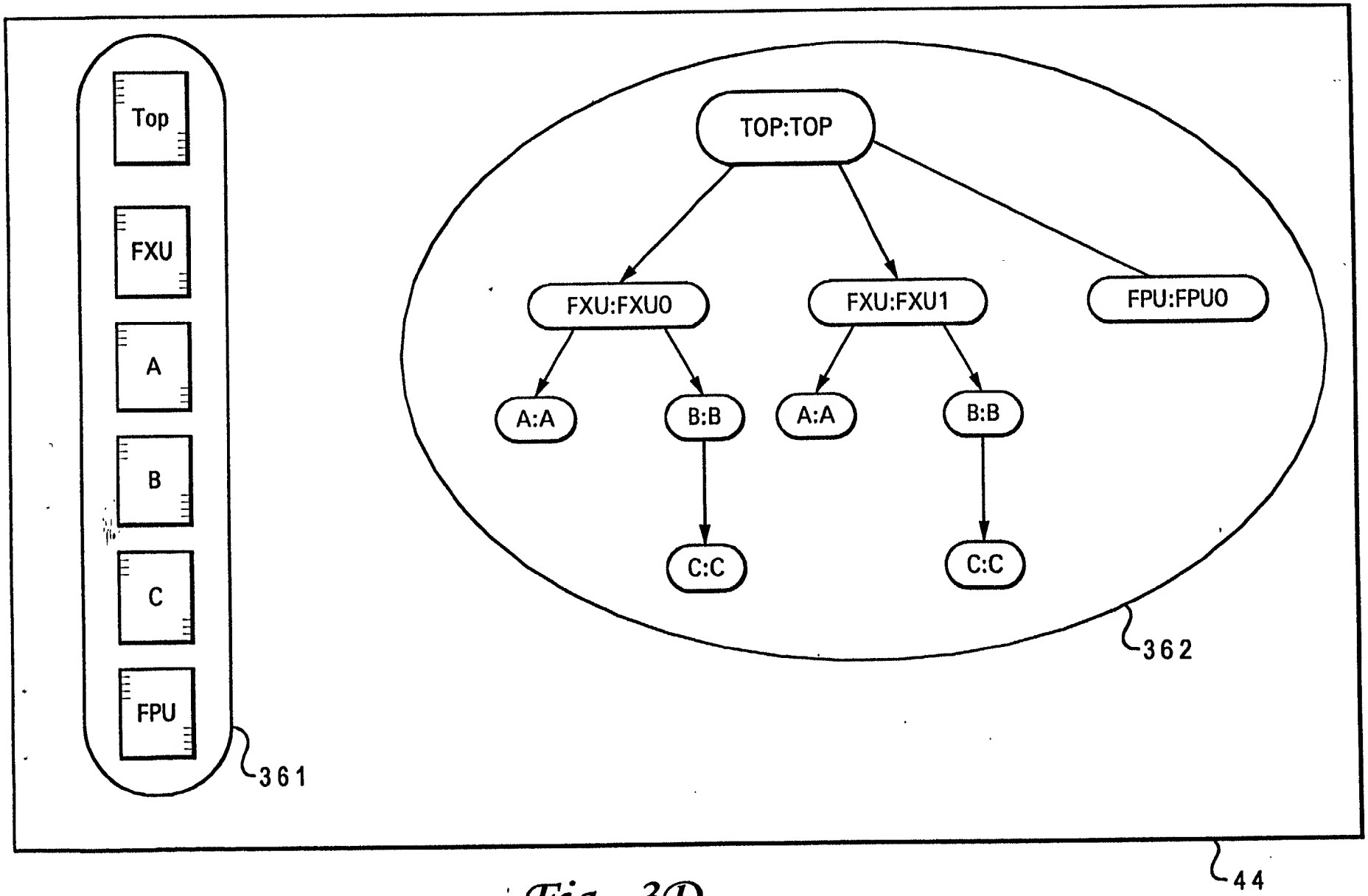

Fig. 3D

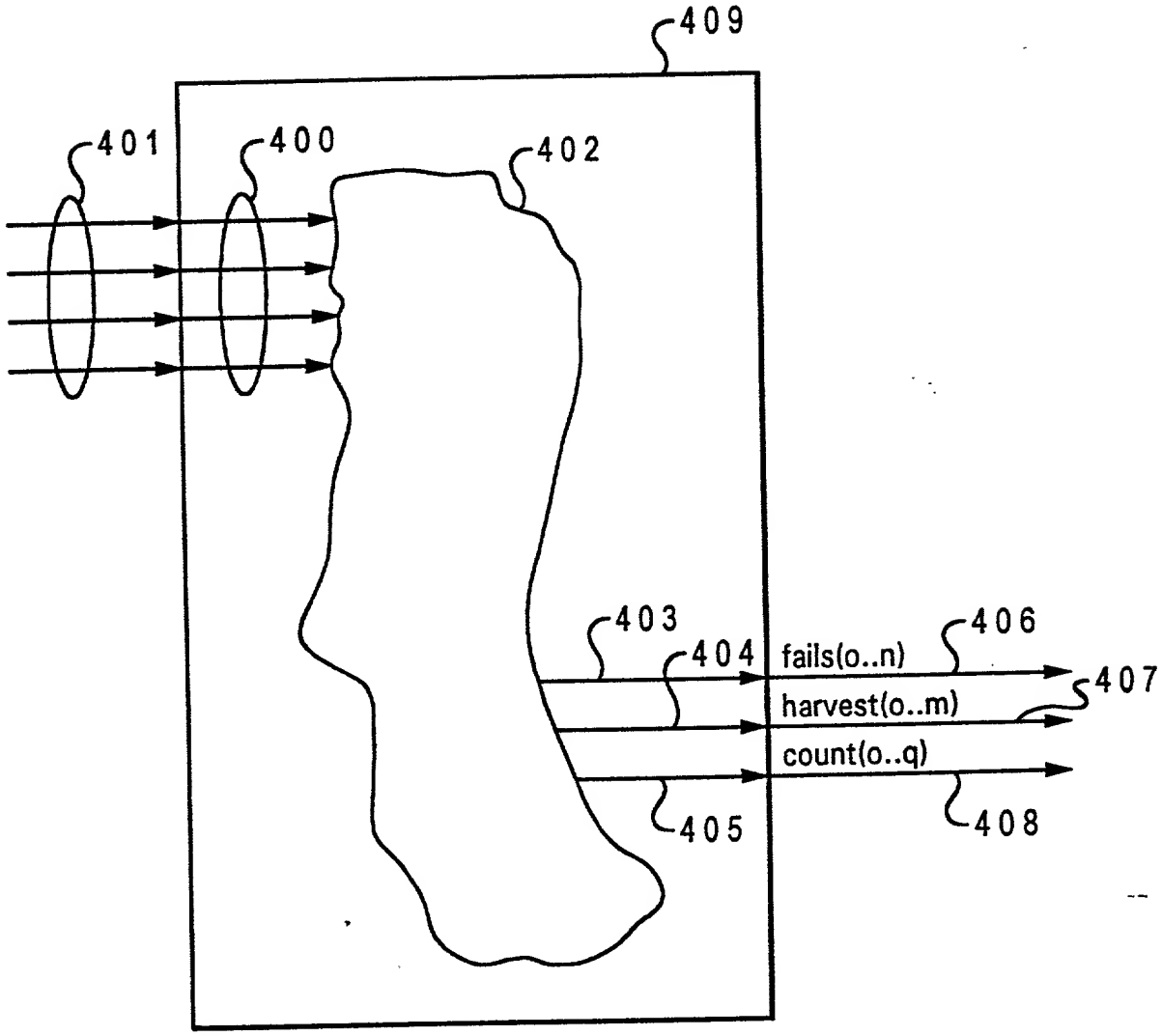
409
401
400
402
403
404
406
fails(o..n)
407
harvest(o..m)
count(o..q)
405
408

Fig. 4A

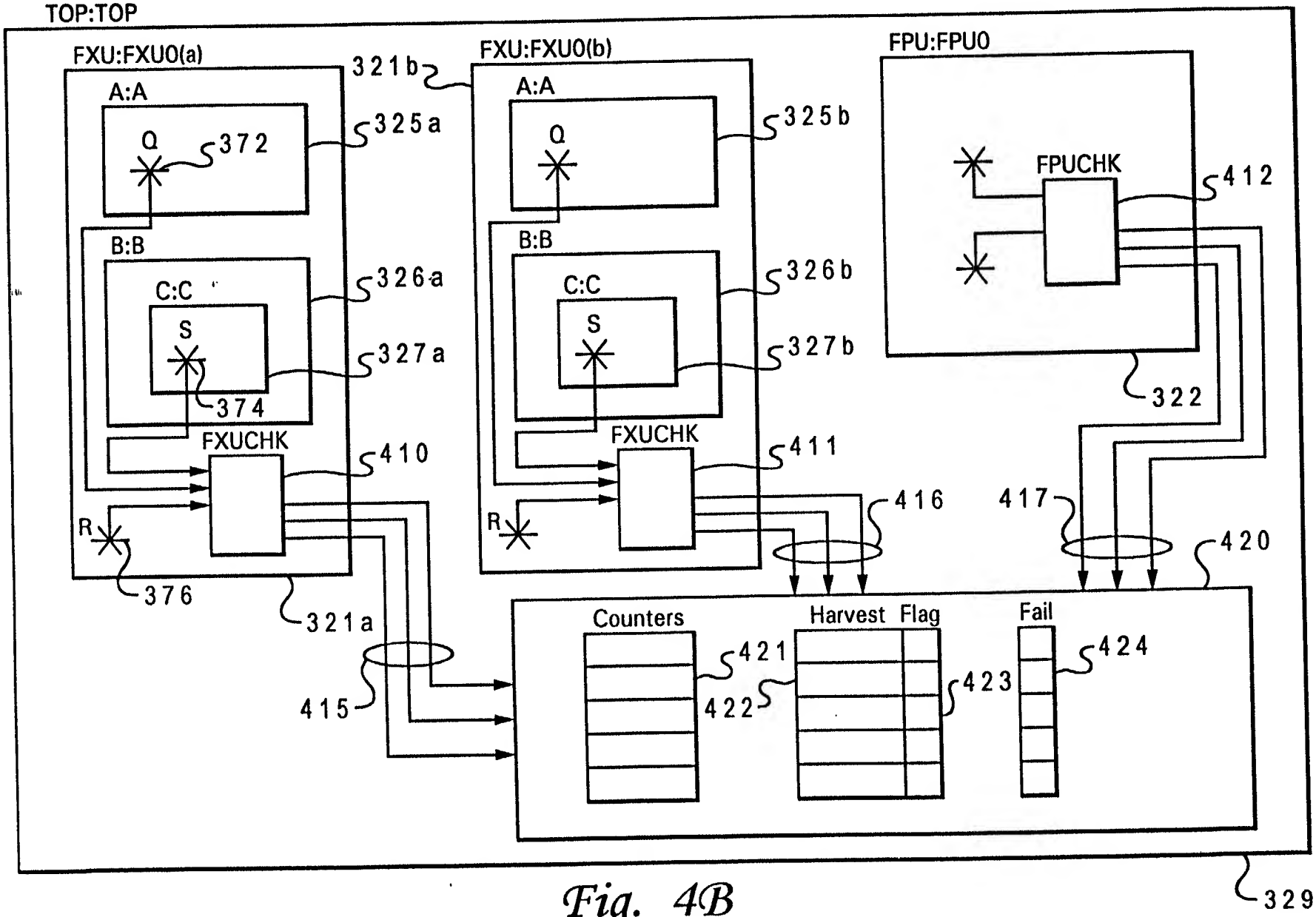

Fig. 4B

```
ENTITY FXUCHK IS

    PORT(   S_IN        :    IN std_ulogic:
            Q_IN        :    IN std_ulogic:
            R_IN        :    IN std_ulogic:
            clock       :    IN std_ulogic:
            fails       :    OUT std_ulogic_vector(0 to 1);
            counts      :    OUT std_ulogic_vector(0 to 2);
            harvests    :    OUT std_ulogic_vector(0 to 1);
        );

--!! BEGIN
--!! Design Entity: FXU;

--!! Inputs
--!! S_IN        =>    B.C.S;
--!! Q_IN        =>    A.Q;
--!! R_IN        =>    R;
--!! CLOCK       =>    clock;
--!! End Inputs

--!! Fail Outputs;
--!! 0 : "Fail message for failure event 0";
--!! 1 : "Fail message for failure event 1";
--!! End Fail Outputs;

--!! Count Outputs;
--!! 0 : <event0> clock;
--!! 1 : <event1> clock;
--!! 2 : <event2> clock;
--!! End Count Outputs;

--!! Harvest Outputs;
--!! 0 : "Message for harvest event 0";
--!! 1 : "Message for harvest event 1";
--!! End Harvest Outputs;

--!! End;

ARCHITECTURE example of FXUCHK IS

BEGIN

    ... HDL code for entity body section ...

END;
```

450 452 453 454 455 456 457 451 458 440

Fig. 4C

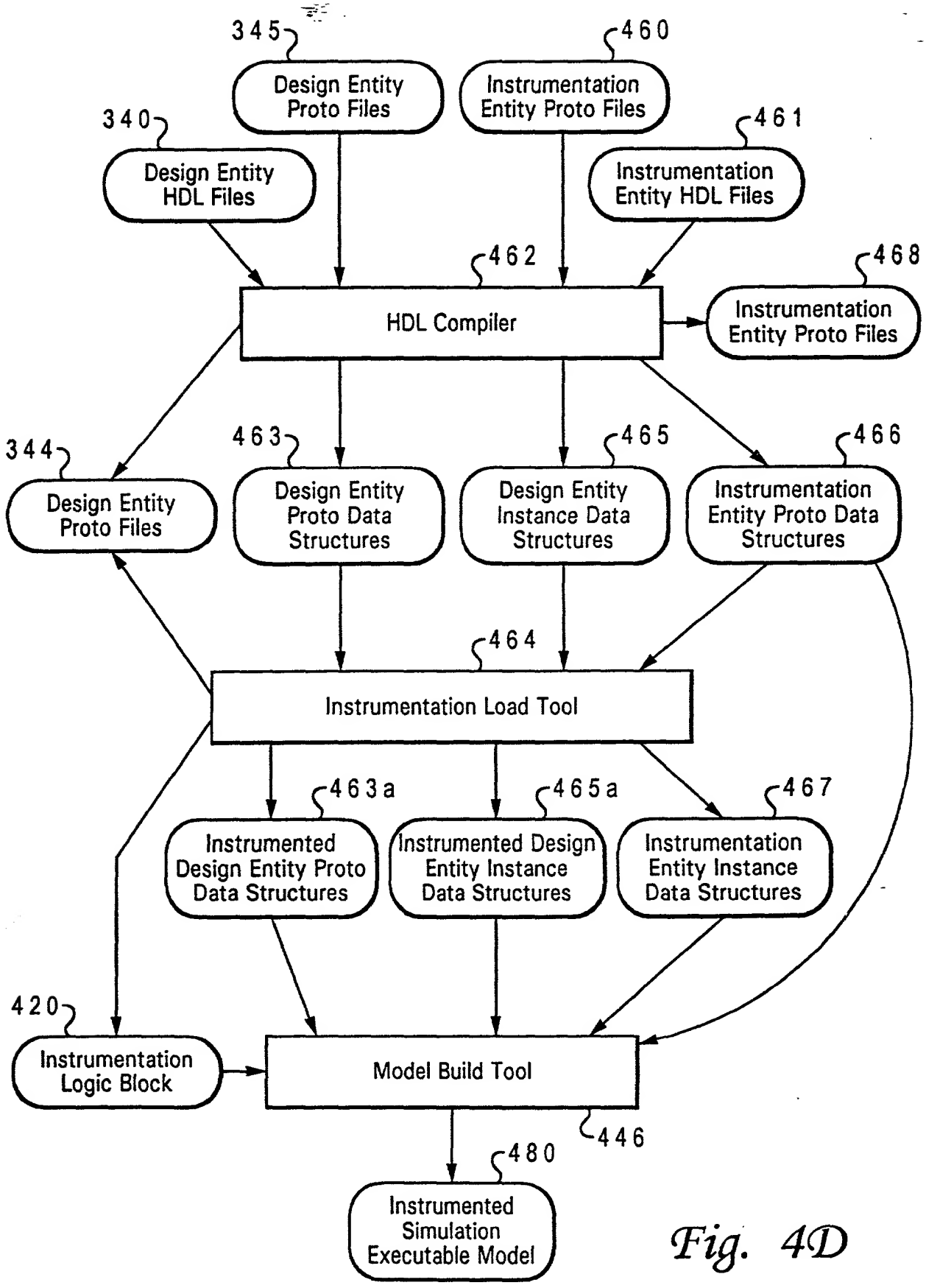
345
Design Entity Proto Files
460
Instrumentation Entity Proto Files
340
Design Entity HDL Files
461
Instrumentation Entity HDL Files
462
HDL Compiler
468
Instrumentation Entity Proto Files
344
Design Entity Proto Files
463
Design Entity Proto Data Structures
465
Design Entity Instance Data Structures
466
Instrumentation Entity Proto Data Structures
464
Instrumentation Load Tool
463a
Instrumented Design Entity Proto Data Structures
465a
Instrumented Design Entity Instance Data Structures
467
Instrumentation Entity Instance Data Structures
420
Instrumentation Logic Block
Model Build Tool
446
480
Instrumented Simulation Executable Model

Fig. 4D

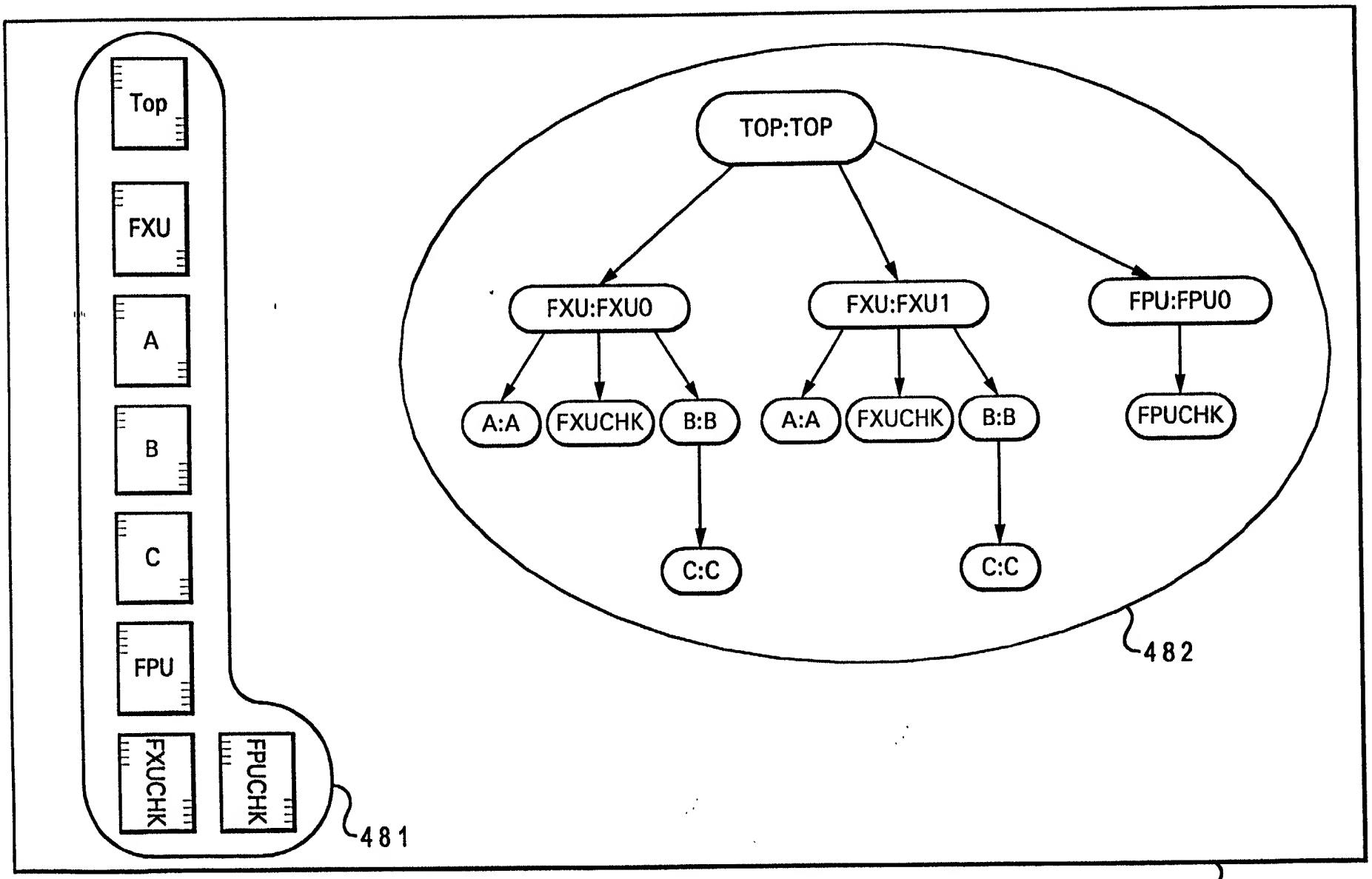
Top
FXU
A
B
C
FPU
FXUCHK
FPUCHK
481
TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
FXUCHK
B:B
C:C
A:A
FXUCHK
B:B
C:C
FPUCHK
482
44

Fig. 4E

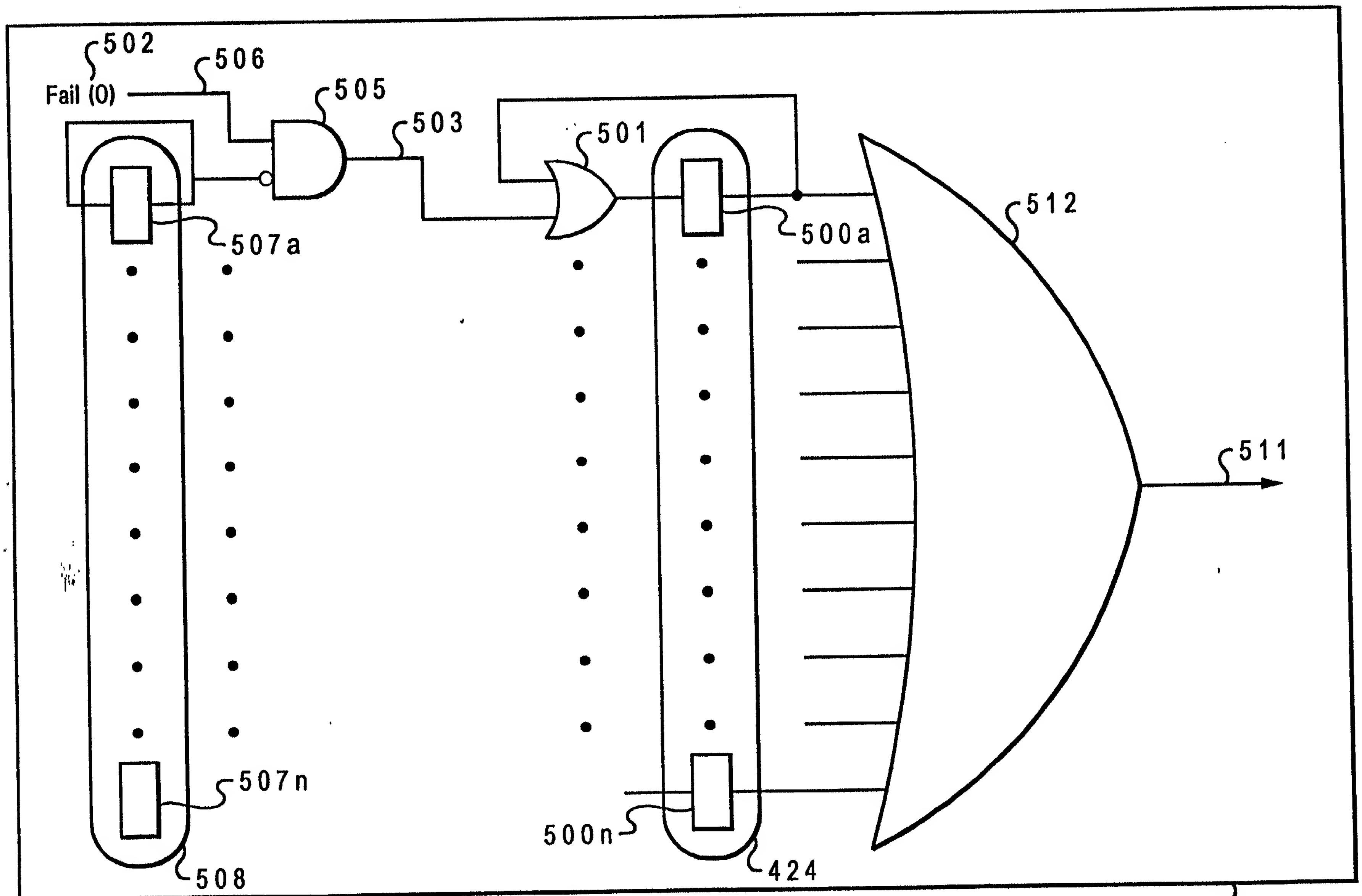
502
Fail (0)
506
505
503
501
507a
500a
512
511
507n
500n
508
424
420

Fig. 5A

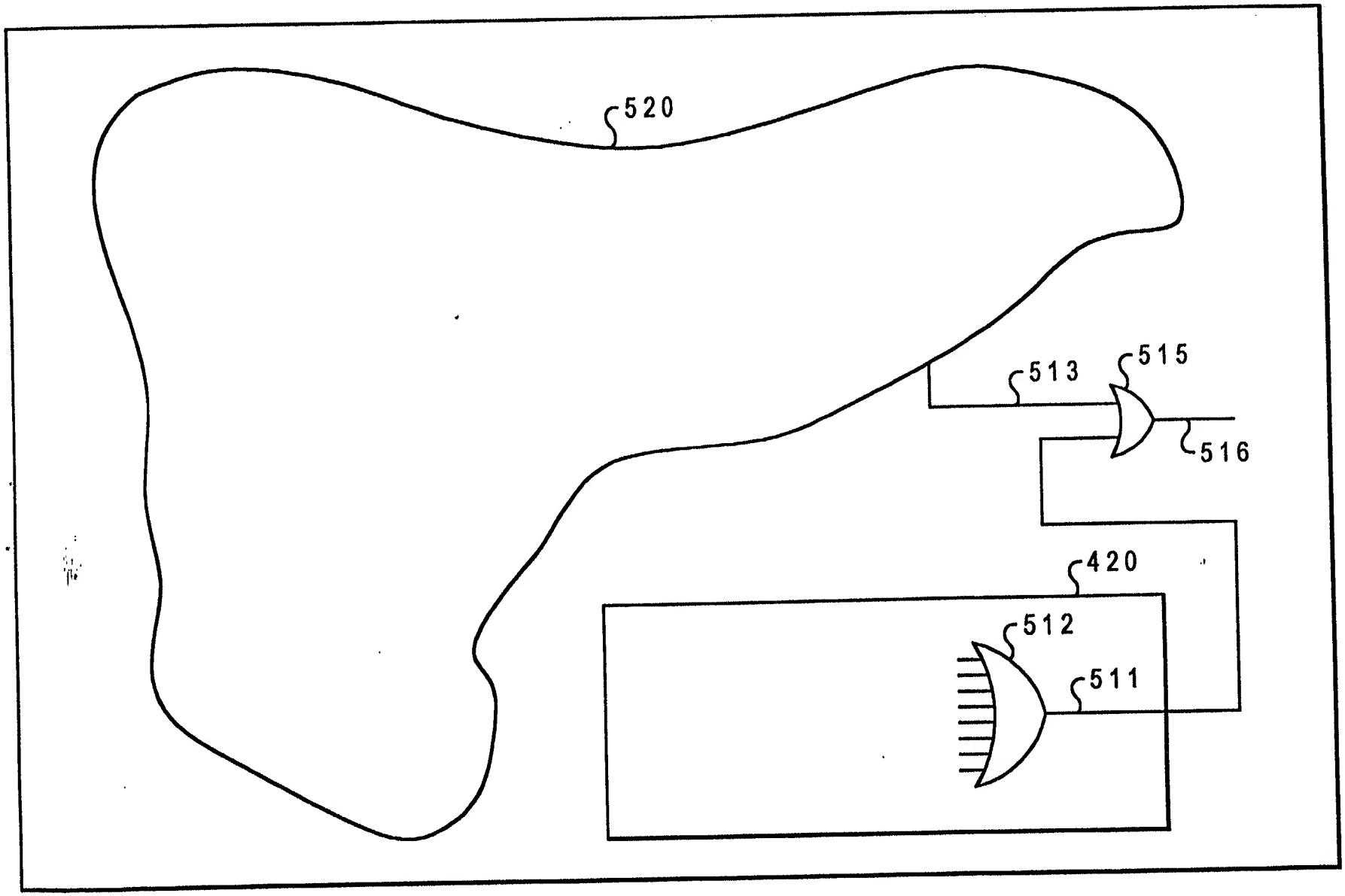

Fig. 5B

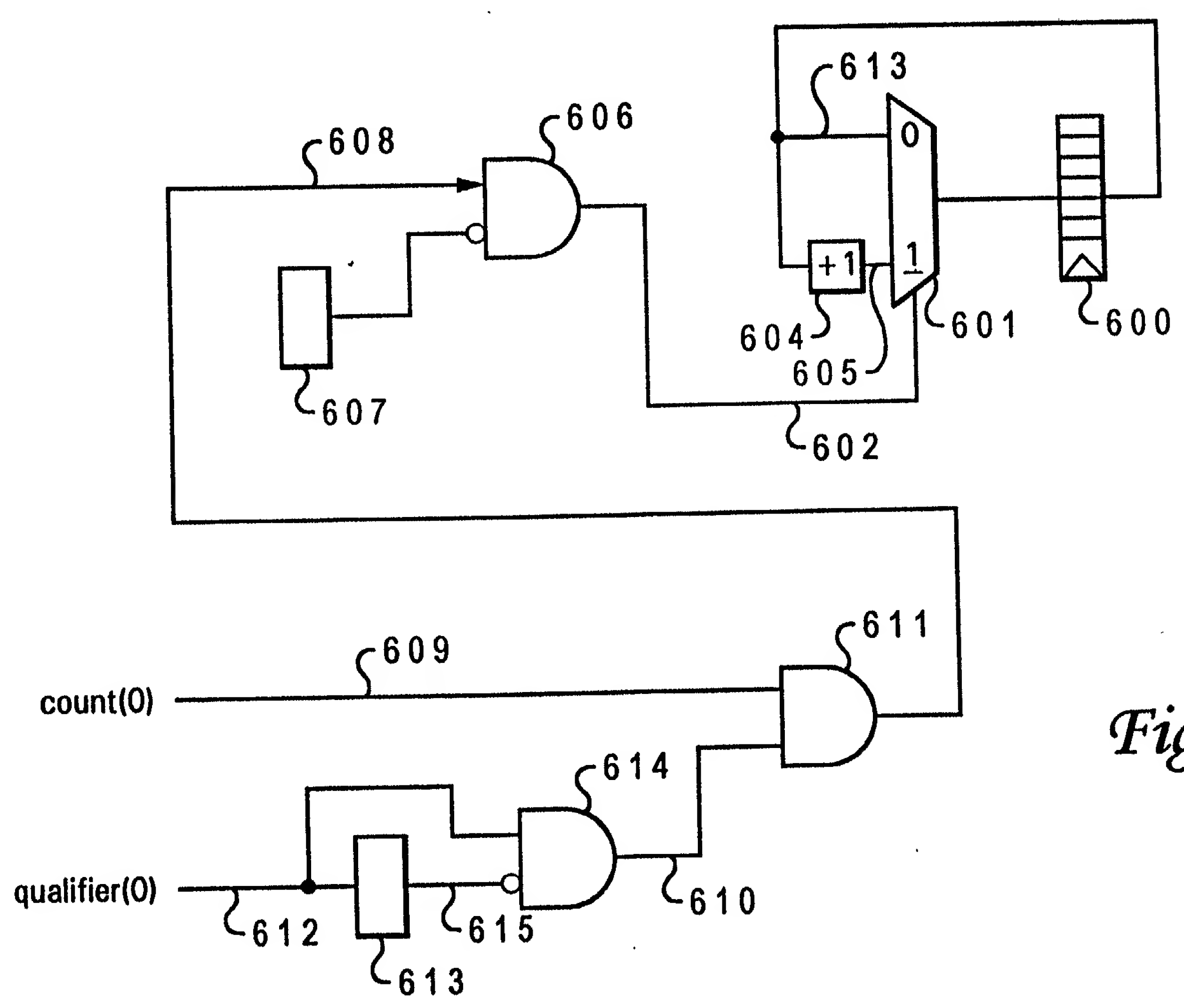

Fig. 6A

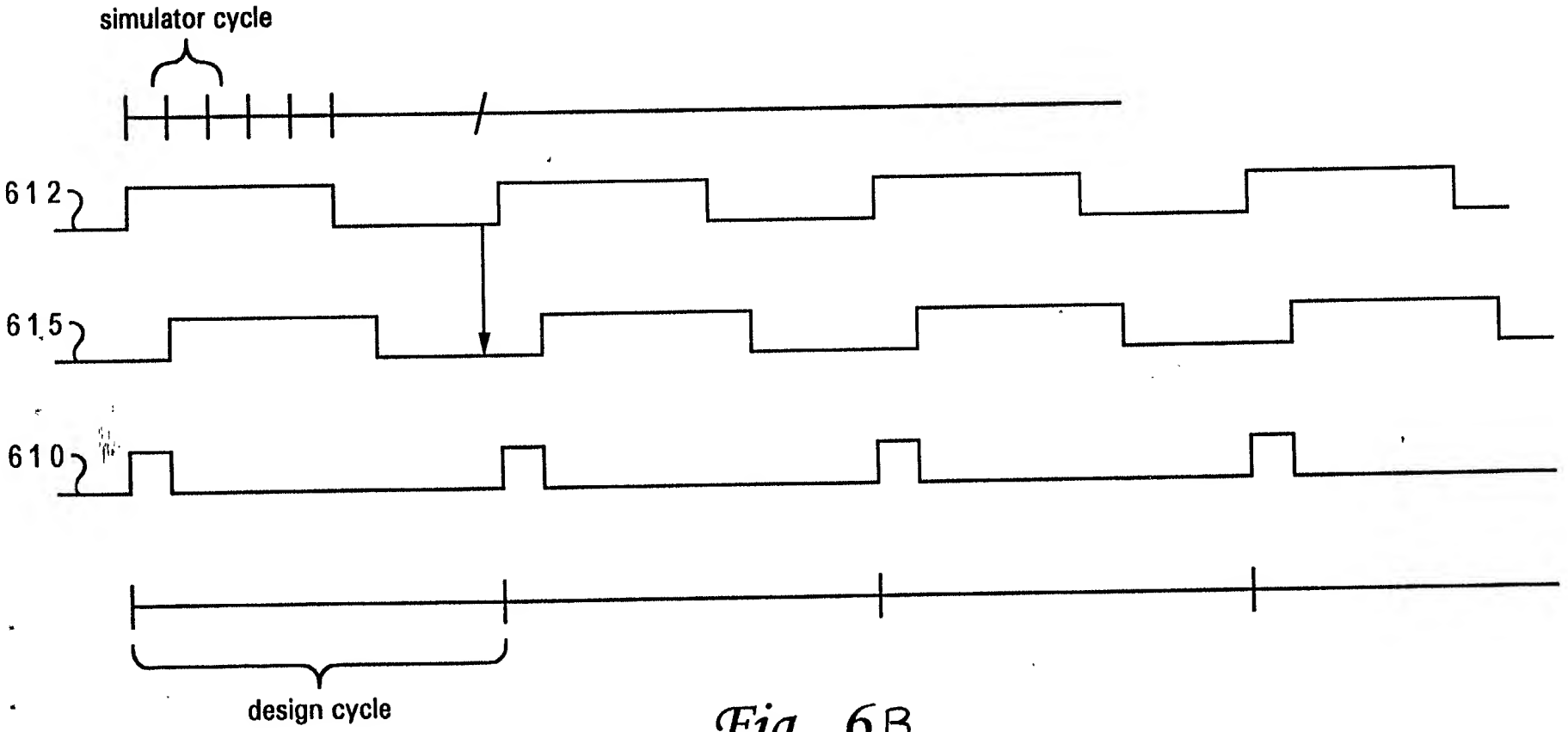

Fig. 6B

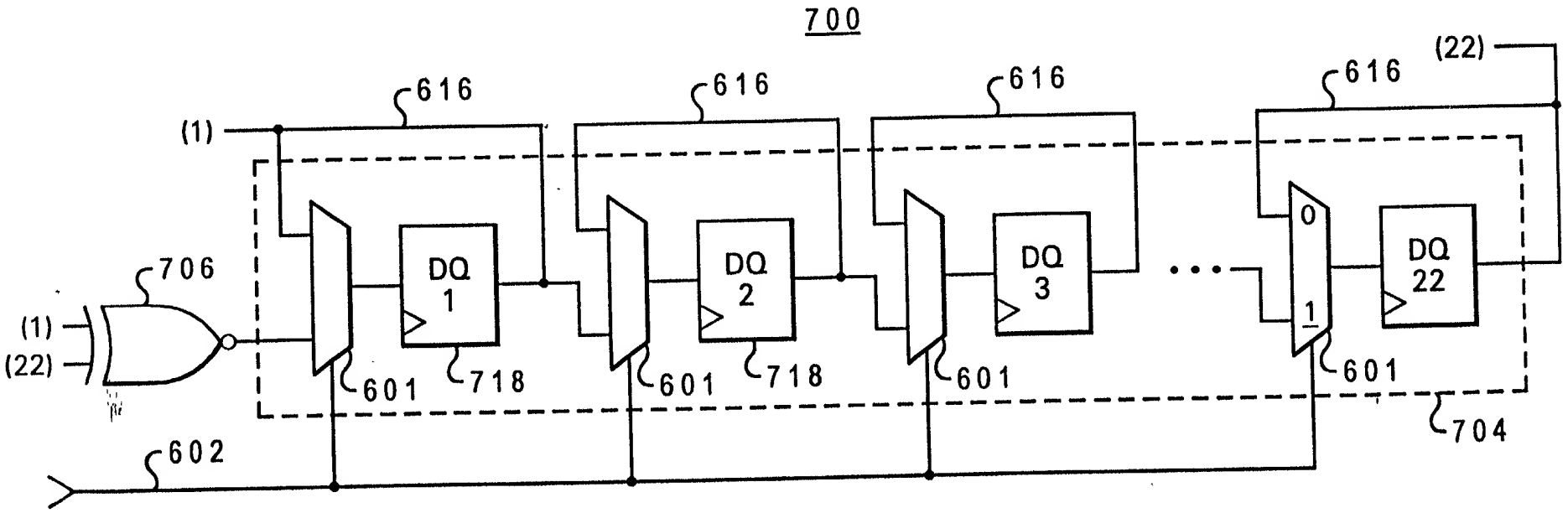

Fig. 7

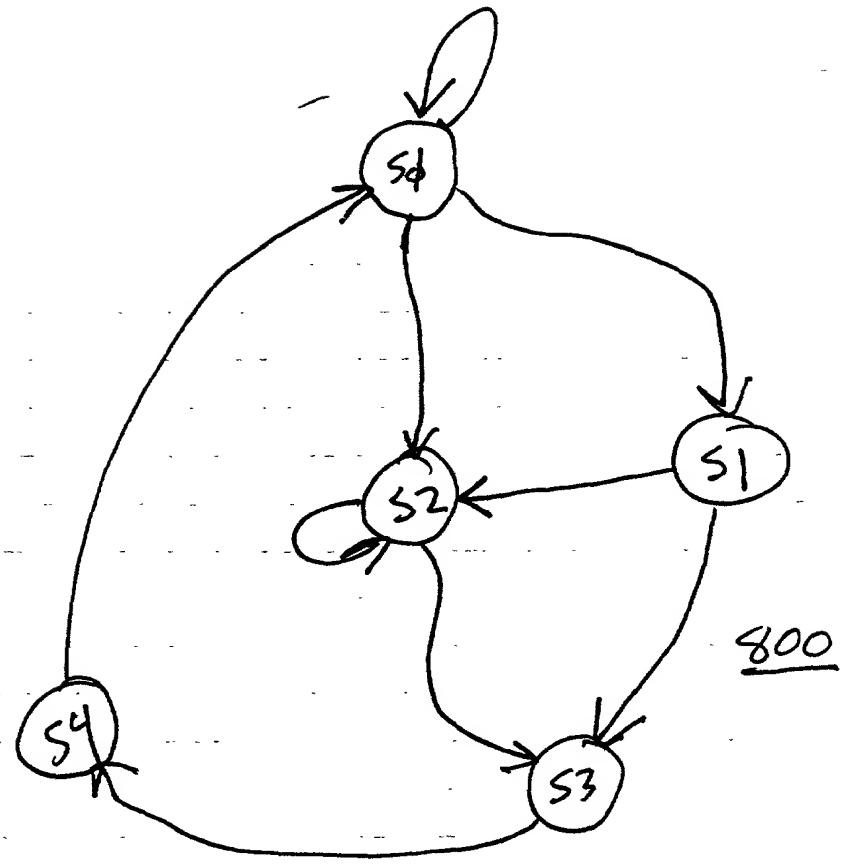

FIG. 8

(Prior Art)

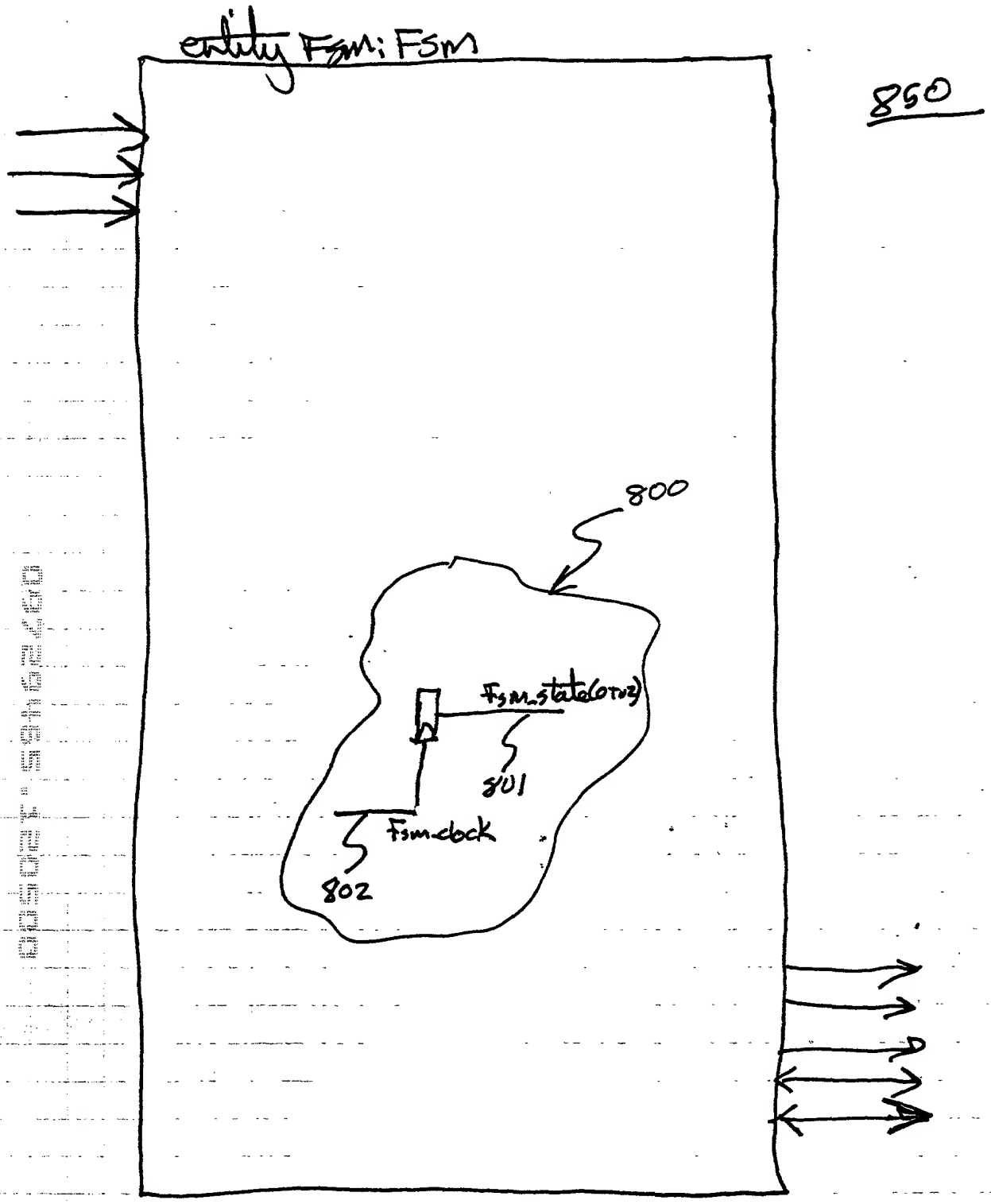

FIG. 8A
(Prior Art)

```
entity Fsm is
  PORT(
        .... ports for entity Fsm.....
  );
ARCHITECTURE Fsm of Fsm IS
BEGIN
        .... HDL code For Fsm and restofthe entity...
        fsm_state(0 to 2) <= ... signal 801 ....
853 {   --!! Embedded Fsm : exampleFsm;
859 {   --!! clock        : (fsm_clock);
854 {   --!! state_vector : (fsm_state(0 to 2));
855 {   --!! states       : (s0,s1,s2,s3,s4);
856 {   --!! state_encoding : ('000','001','010','011','100');
857 {   --!! arcs         : (s0=>s0, s0=>s1, s0=>s2,
        --!!                s1=>s2, s1=>s3, s2=>s2,
        --!!                s2=>s3, s3=>s4, s4=>s0);
858 {   --!! end Fsm;
END;
```

852

86

FIG. 8B

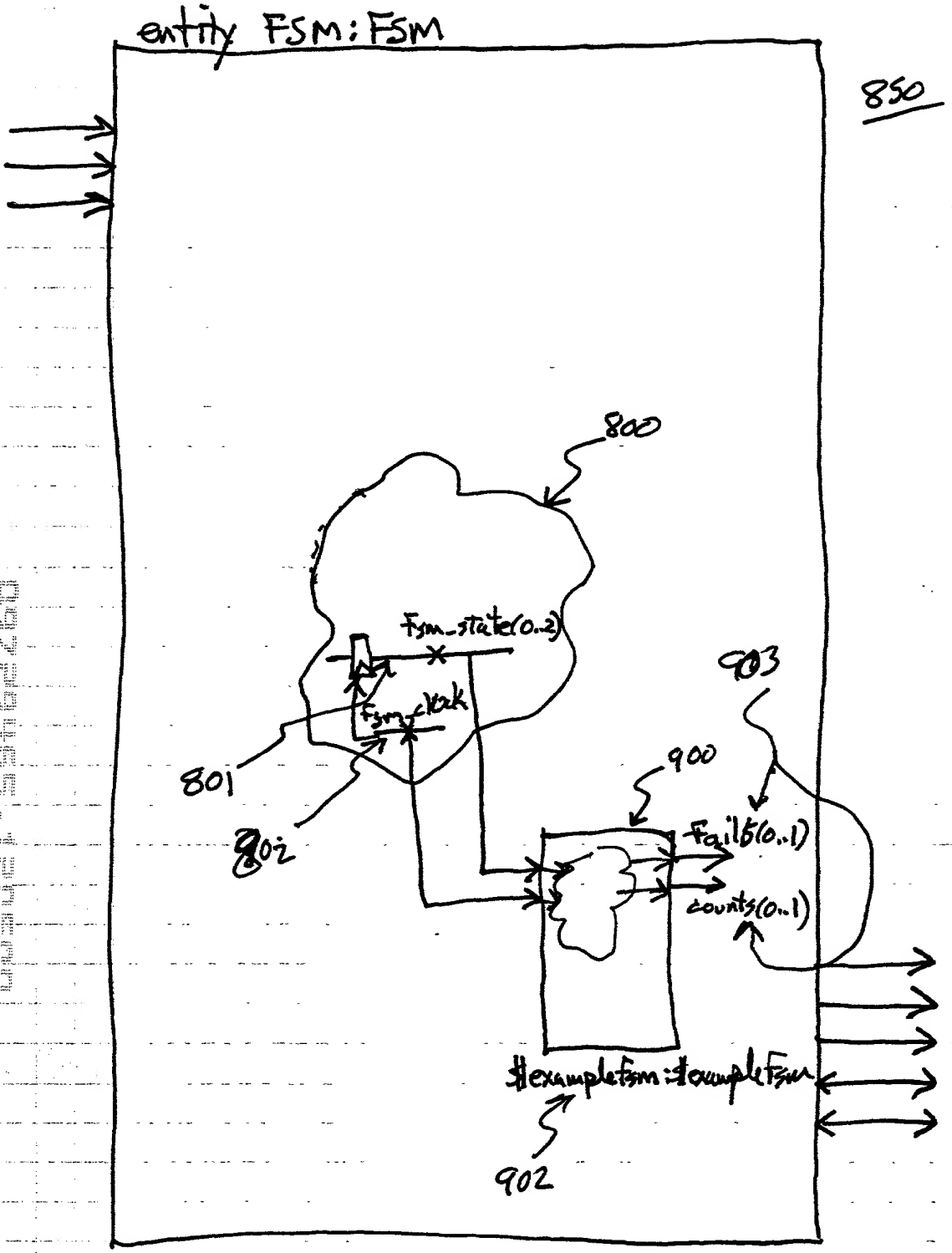
entity FSM:FSM
850
800
fsm_state(0..2)
fsm_clock
801
802
903
900
Failb(0..1)
counts(0..1)
exampleFsm:exampleFsm
902

FIG. 9